Quantum Machine Learning in Finance for Continuous Variable Price Prediction

Team: Pratjz

Outline

- Objective
- QML (Quantum Machine Learning)
- QNN (Quantum Neural Networks)
- Pennylane QML Library
- Implementation Details
- Results
- Conclusion & Future Research

Objective

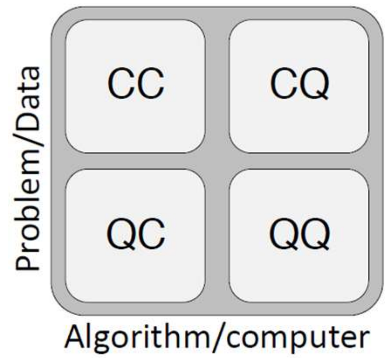
- The aim of this study is to develop a clear understanding of the promises and limitations of the current quantum algorithms for machine learning. And then draw the parallels to recent ‘quantum-inspired’ results, and explain the implications of these results for quantum machine learning applications.

- In this work, we will use **Quantum Neural Network** to present quantum algorithms for financial applications, focusing on Continuous variable prediction problems for example **Asset price prediction**.

- We will use **Boston housing** data to predict property prices, Dataset is taken from StatLib library maintained by Carnegie Mellon University.

QML (Quantum Machine Learning a Brief Introduction)

- Quantum machine learning can generally be divided into following four distinct areas

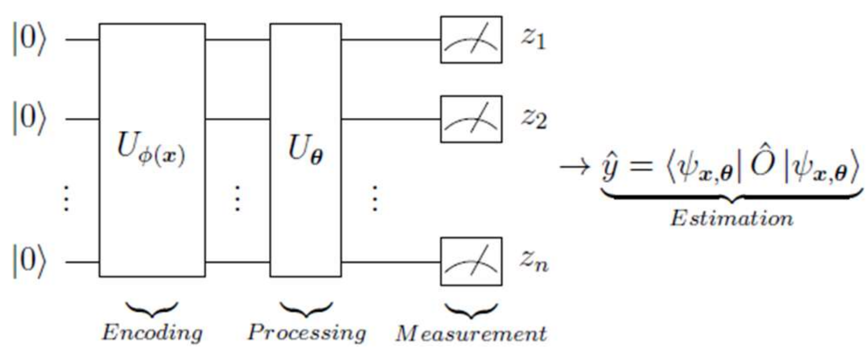


- CC: ML algorithms that are executed on classical computers and applied to classical data.
- **QC**: ML algorithms that are executed on quantum computers and applied to classical data.
- CQ: ML algorithms that are executed on classical computers and applied to quantum data.
- QQ: ML algorithms that are executed on quantum computers and applied to quantum data.

The biggest attention is being paid to quantum algorithms that perform machine learning on classical data **QC** due to growing Complexity and size of data Silos. There are many QML techniques…but in this study we focus on QNNs.

QNN (Quantum Neural network)

- *Parameterized quantum circuits* (**PQC**) more commonly termed as Quantum Neural networks (**QNN)**, are quantum circuits comprised of fixed gates, such as CNOT gates, and adjustable gates, such as Pauli rotations.
- QNNs are general algorithms with free parameters that needs to be adjusted in order to solve a given problem. Quantum computers are used to evaluate the circuits, while classical hardware is used to post-process the results and optimize the parameters.
- This hybrid approach is much less demanding on the number of qubits and the depth of the circuit. Thus, they are much more suitable for NISQ era.
- Typical structure of QNNs can be broken into three stages: *feature encoding*, *processing* and *measurement* with potential post-processing as summarized in following figure

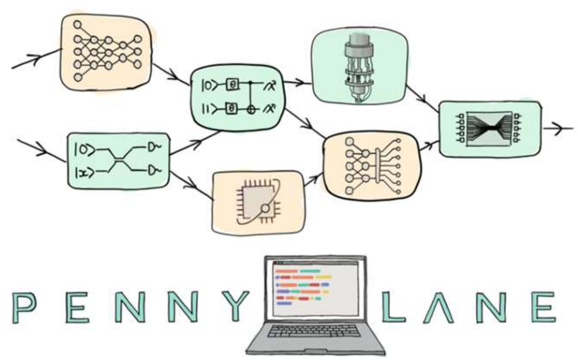


Pennylane QML Library

- Pennylane is an open-source Quantum ML framework by Xanadu, built in Python to achieve machine learning tasks with quantum computers.

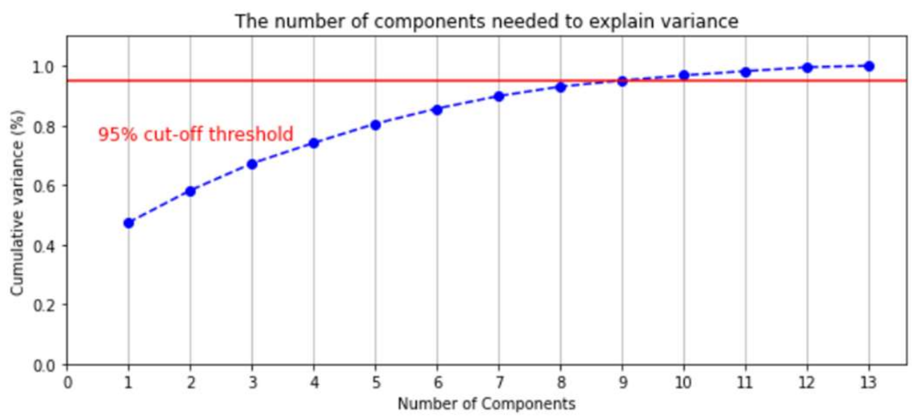
- It Supports hybrid quantum and classical programming allowing users to connect quantum hardware with various other frameworks like PyTorch, TensorFlow, Qiskit, Cirq etc..

- It Is Hardware agnostic i.e. same quantum circuit model can be run on different backends—and allows plugins for access to diverse devices, including Photonic Strawberry Fields, Amazon Braket, IBM Q, Google Cirq, Rigetti Forest, Microsoft QDK and ProjectQ



Implementation Details

- **Data Set** – Boston Housing data contains attributes of houses at different locations around the Boston suburbs in the late 1970s. Target variable is the median price value of the houses (in k$). The data has total 506 samples and 13 features.
- We split the data into Train Test datasets with 70% train & 30% test data
- We reduce the number of features and determine number of qubits desired using Principal component analysis

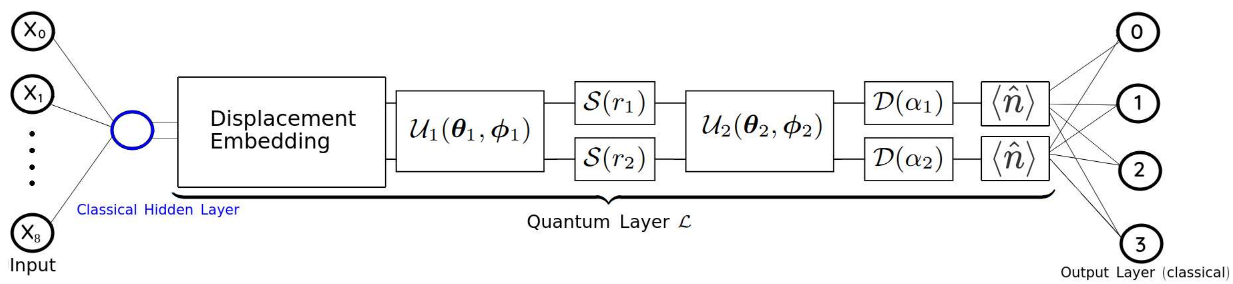


- We can see that to get about 95% of variance explained we need 9 principal components which will also be the number of Qubits we will use for our Quantum Circuit.

Implementation Details

Quantum Neural Network

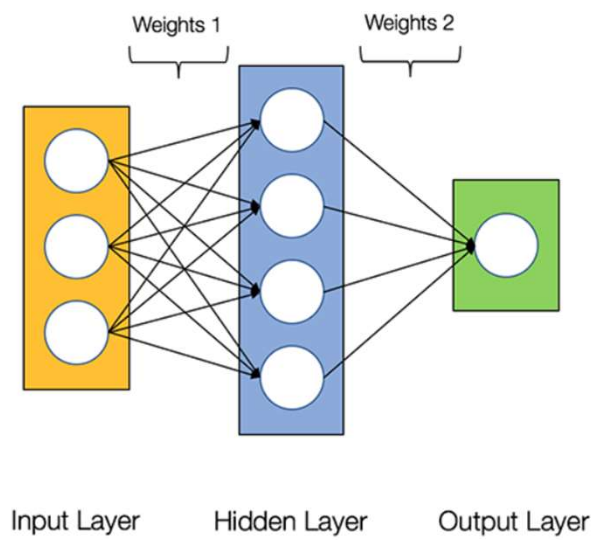
- First we map our data features to the qubits using *AngleEmbedding* Template from pennylane. It encodes N features into the rotation angles of n qubits. We choose *AngleEmbedding* over *AmplitutedEmbedding* because the dataset is fairly large and same is the number of features which would be very complex and thus not possible on current QPUs to encode using amplitude encoding.
- For the QNN layer we use *StronglyEntanglingLayers* circuit from pennylane templates library. It allows us to train Quantum Layer using features as angle parameters and is better suited for predictions involving continuous variables.
- The QNN is built using three layers two classical layers in tensorflow using pennylane wrapper and one Quantum layer, wherein the QNN layer is sandwiched between the two Input and output classical layers.We use the Stochastic Gradient Descent Optimizer for the optimisation step



Implementation Details

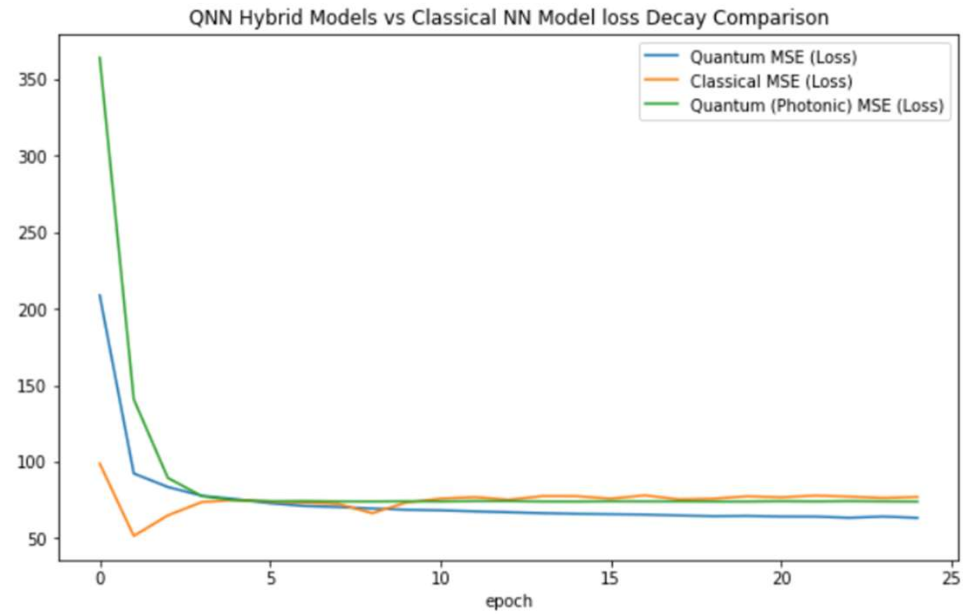
Classical Neural Network

- For comparison we also create a simple Classical Neural Network using Tensorflow and Keras
- The architecture of the NN contains three layers similar to our earlier QNN, the input layer has number of features as inputs and the output layer has one output for the Price variable
- We use the Same Stochastic Gradient Descent Optimizer



Results

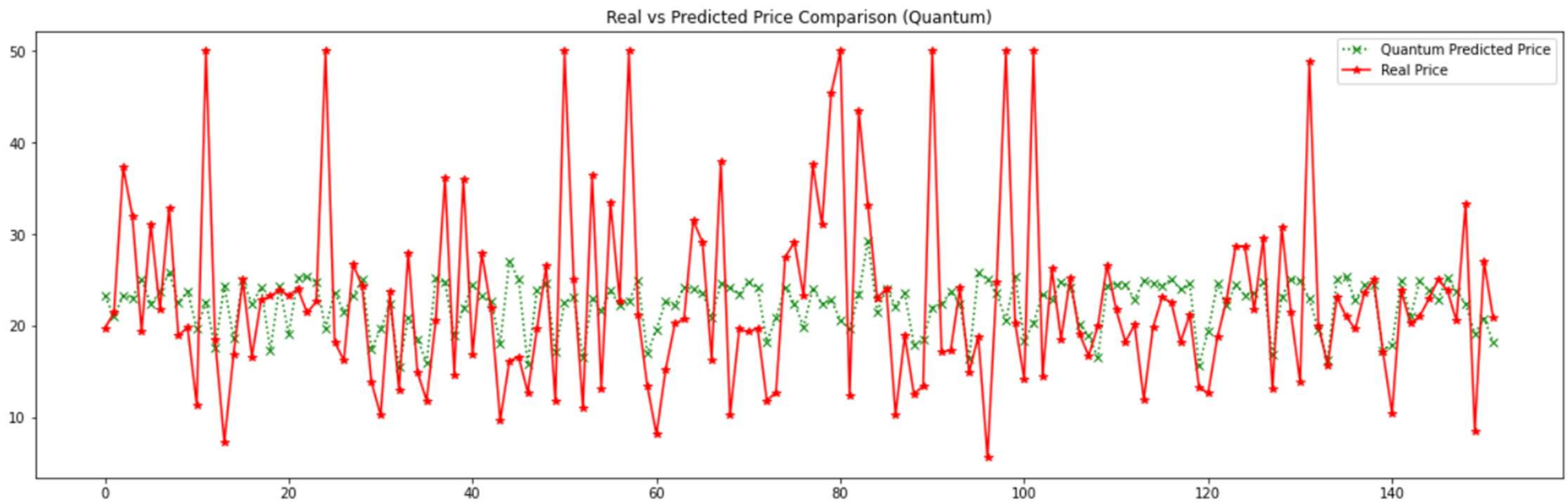
Following Figure illustrates the loss decay comparison with number of epochs, lower the loss higher is the prediction accuracy of the model.



We can see that *<u>Quantum Neural network model performs slightly better than the simple classical neural network</u>* with the same settings & configurations.

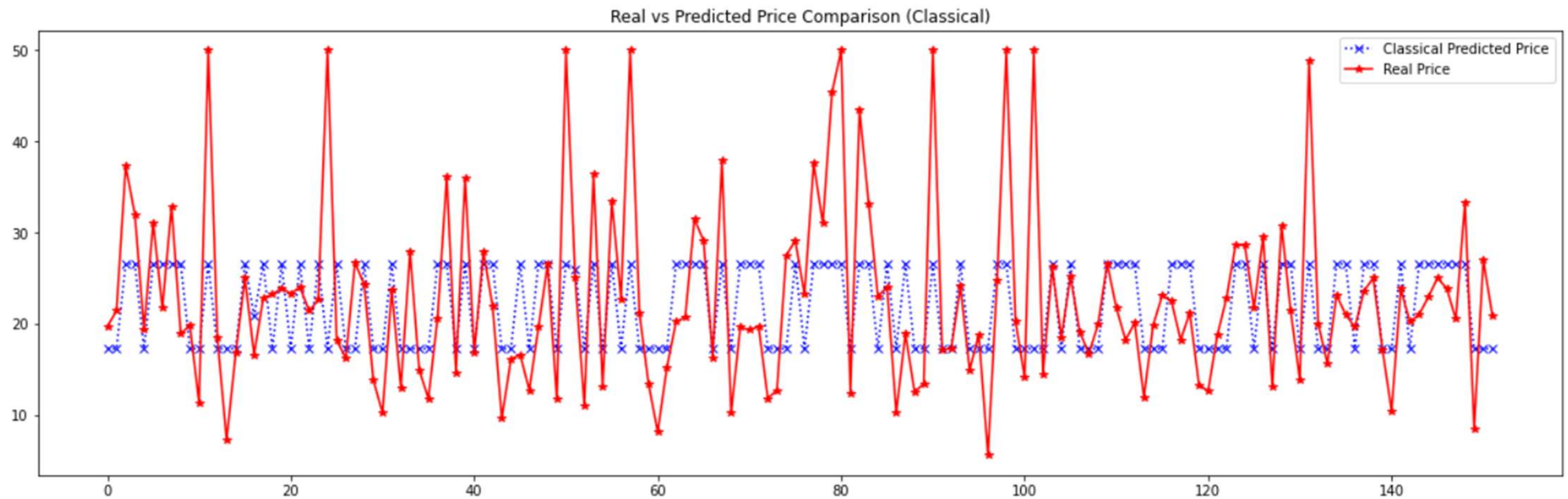
Results

Comparing Prices predicted by Quantum NN model to Actual Prices: Here we can see that the model has ***generalized fairly well*** and the predicted prices are *closer* to the actual prices, this could be much better with more complex larger QNN architectures.



Results

Comparing Prices predicted by Classical NN model to Actual Prices: Here we can see that the model has *not generalized* as well as the QNN model, Of course the model architecture could be made much better to make better predictions but with comparable settings to the QNN, The QNN generalizes better.



Conclusion & Future Research

From the results and comparisons with fair confidence we can deduce that

- Even with the simplest settings and configurations the Quantum Neural networks *gives little better accuracy & generalizes fairly well* compared to the classical NN and as the technology & research matures, quantum hybrid approaches could show a promising advantage over classical counterparts for plenty of problems.
- We can use quantum machine learning for feature spaces that have very high dimensionality which are computationally very costly to compute classically especially in areas where there is involvement of classically interactable problems for example finance & quantum chemistry.

Future Research

- Explore Similar Quantum Techniques on other areas in Finance & quantum chemistry.
- Try different Photonic Continuous Variable QNN Ansatz described in newer research papers
- Currently I am actively contributing to Pennylane library building new features & fixing issues, would like to create Finance specific API using pennylane.

References

- Maria Schuld and Francesco Petruccione. Supervised Learning with Quantum Computers.
- Boston Housing – Price Prediction https://rstudio-
- Pennylane.ai documentation https://pennylane.readthedocs.io/en/stable/
- Quantum Machine Learning for Finance https://arxiv.org/pdf/2109.04298.pdf
- Continuous-variable quantum neural networks https://arxiv.org/abs/1806.0687

Few Probable Application Areas of QML in Finance

Here we list some areas in financial space that can be good candidate to explore any quantum advantage using Quantum Machine learning techniques

- **Regression techniques using QML:**
  - Asset Pricing
  - Multi-Asset Trend Following Strategy
  - Implied Volatility Estimation
- **Classification techniques using QML:**
  - Binary Options Reduction
  - Financial Forecasting
  - Credit Scoring
- **Clustering techniques using QML:**
  - Fraud Detection
  - Stock Selection
  - Exchange Rate Regimes
  - Hedge Fund Clustering
- **Generative Modelling techniques using QML:**
  - Probability Distribution Preparation
- **Quantum Assisted Feature Extraction techniques using QML:**
  - Model Reduction
  - Combinatorial Feature Selection for Credit Score Classification
- **Reinforcement Learning techniques using QML:**
  - Algorithmic Trading

All the techniques are outlined in more detail here[13]